electronic design

Power Gating: Reducing PCIe Power Consumption to Mobile Levels

Electronic Design
Richard Solomon

Thu, 2015-11-05 15:00

PCI Express has long been the dominant I/O interconnect in traditional PCs and servers. It has even made its way into laptops and notebooks. Only recently, though, has PCI Express been found in smaller mobile devices like smartphones and tablets. For years, the industry perception was that PCI Express was too "power hungry" to ever succeed in such smaller battery-powered devices, so what has changed?

This article explores two technological advances that helped dramatically reduce the power consumption of the PCI Express interface—one a protocol enhancement, the other a new silicon technique.

New Protocol, Lower Power

To understand PCI Express' new power-saving protocol, we need to look back at history. During the evolution of the original parallel PCI bus, around 1997, power-saving "Device States" or "D-States" were introduced with "D0" reflecting the normal full-power operation of a device, "D3" indicating a device either powered down or ready to do so, and optional "D1" and "D2" states for intermediate power savings.

Related

What's the Difference Between PCI Express and Ethernet in Cost and Power Consumption?

In The End, Will It Be PCI Express, USB, And Ethernet?

What's the Difference Between mPCIe and mSATA?

All of those power-saving states were for a device that was not being used. Entering and exiting the power-saving (non-"D0") states required software and operating-system intervention. This specification was fine for operations like putting a laptop into a sleep or hibernation state, where no user interaction was expected for minutes or hours.

By the time PCI Express was developed in 2002, additional "Link States," or "L-States," were included in the specification. These follow the template of the existing "D-States." "L0" reflects a PCI Express link in full operation. "L1" is a link that's not transferring data, but which can relatively quickly resume normal operation. "L2" and "L3" each reflect a link with main power removed ("L2" indicates an auxiliary power supply is active to provide "keep alive" power to devices). An "L0s" state was also defined in which each direction of the PCI Express link could be shut down independently with a quick resumption to normal operation. Fully active devices in "D0" can transition between "L0," "L0s," and "L1" with no software intervention, thus saving power on their own initiative without any operating-system interaction.

ile this seemed like a solution to the problem of active device power savings, the Achilles heel of the States" turned out to be the speed of powering up or powering down. The PCI Express specification called for devices to exit from "L0s" in less than 1 µs, and from "L1" somewhere on the order of 2 to 4 µs. Although PHY designers could idle their receiver and transmitter logic in L1 to meet those resumption times, they were forced to keep their power-hungry common-mode voltage keepers and phase-locked loops (PLLs) powered on and running. This meant that each lane of a PCIe PHY in "L1" could still be consuming 20 to 30 mW of power, which would clearly be too high for a battery-powered device.

By 2012, it was becoming clear that combinations of specialized hardware and software in such mobile devices could handle transitioning PCI Express components between normal and low-power states if only they had a mechanism to do so. The "L1 PM sub-states with CLKREQ" ECN to PCI Express (often referred to simply as "L1 sub-states") was introduced to allow PCI Express devices to enter even deeper power-savings states ("L1.1" and "L1.2"), while still appearing to legacy software to be in the "L1" state.

The key to L1 sub-states is providing a digital signal ("CLKREQ#") for PHYs to use to wake up and resume normal operation. This permits PCI Express PHYs in the new sub-states to completely power off their receiver and transmitter logic, since it's no longer needed for detection or signaling of link resumption. They can also power off their PLLs and potentially even common-mode voltage keepers as the ECN specifies two levels of resumption latency.

The L1.1 sub-state is intended for resumption times of 20 µs (5 to 10 times longer than the L1 sub-state allowed), while the L1.2 sub-state targets times of 100 µs (up to 50 times longer). Both sub-states should permit good PHY designs to power off their PLLs. The L1.1 sub-state requires maintaining common-mode voltage, while the L1.2 sub-state allows it to be released. Well-designed PCI Express PHYs in the L1.1 sub-state should be able to reach power levels around 1/100 of that in L1 state. Likewise, in L1.2 sub-states, those PHYs should reduce power to about 1/1000 of L1 state.

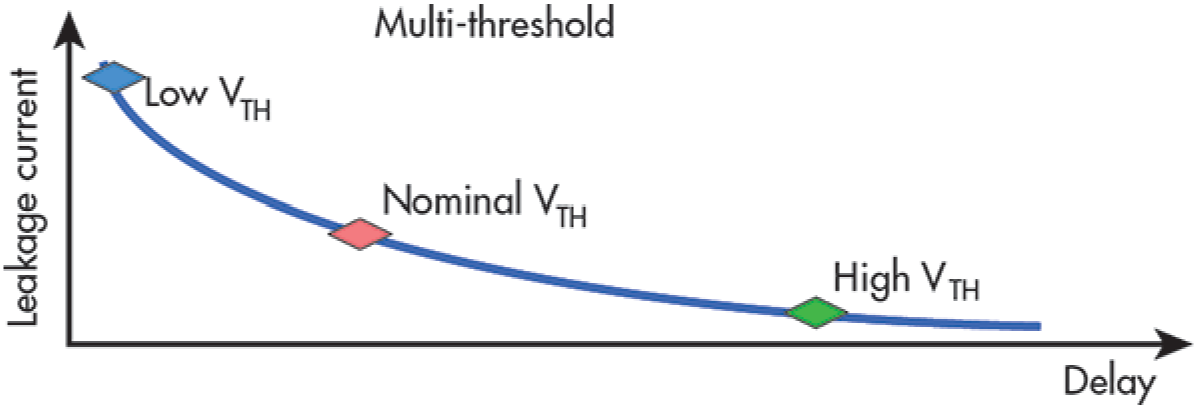
Smaller Process Geometries + Clock Gating to Cut Power

Much has been made of the dramatic decreases in process geometry with traditional gate CMOS technologies down under 30 nm, and FinFET processes quickly headed under 10 nm. SoC designers have benefitted from tremendous gate-count increases, allowing for more complex logic to be packed into cutting-edge designs. Raw speed, in the form of decreasing gate delays and the corresponding increases in clock frequency, has followed shrinking geometries, albeit at a slower rate.

When discussing power consumption, the focus has traditionally been on dynamic current—the power consumed each time a CMOS device changes state—as happens each clock cycle. Higher clock frequencies naturally result in more power consumed per unit of time; therefore, designers can tune their power usage in direct proportion to their device's operating frequencies.

With PCI Express PHY idle power down to a level that's small-device-friendly, the industry's attention turned to digital logic. In a nod to power consumption, some SoC designers have manually implemented clock-gating techniques to stop wasting power in unused portions of the SoC. Consider, for example, a dual-port device with one port unconnected. Substantial power could be saved by stopping the clock to that unused port logic and reducing the dynamic power to nothing. Unfortunately, even when a CMOS device is held in a static state, it's consuming some amount of power, known as leakage current. For a PCI Express endpoint controller that implements four lanes of PCIe 3.1's 8GT/s ("Gen3") and might commonly be found in a PCI Express SSD, that leakage could result in 1 to 2 mW of power in the digital-controller logic.

Smaller silicon geometries have improved dynamic power by virtue of their smaller and smaller transistors, but leakage power has not been improving at anywhere near the same rate. Thus, when considered in proportion to ıamic power, the impact of leakage power has actually become quite significant.



To assist in this, process designers have begun offering a range of cells in each library, offering tradeoffs of speed versus leakage power. As shown in Figure 1, low V_T cells are fast but have high leakage current, and while high V_T cells have low leakage current, they're also much slower. Today's popular FinFET processes do offer lower leakage than past traditional gate silicon processes. But even in FinFET processes, the ultra-low V_T cells needed for gigahertz operation may still exceed the required power targets for mobile devices.

Design implementation tools are now starting to offer automated ways to assist with the selection of transistor threshold, and even assist with or automate the insertion of clock-gating logic *(Fig. 2)*. This allows more unused logic to consume only leakage current.

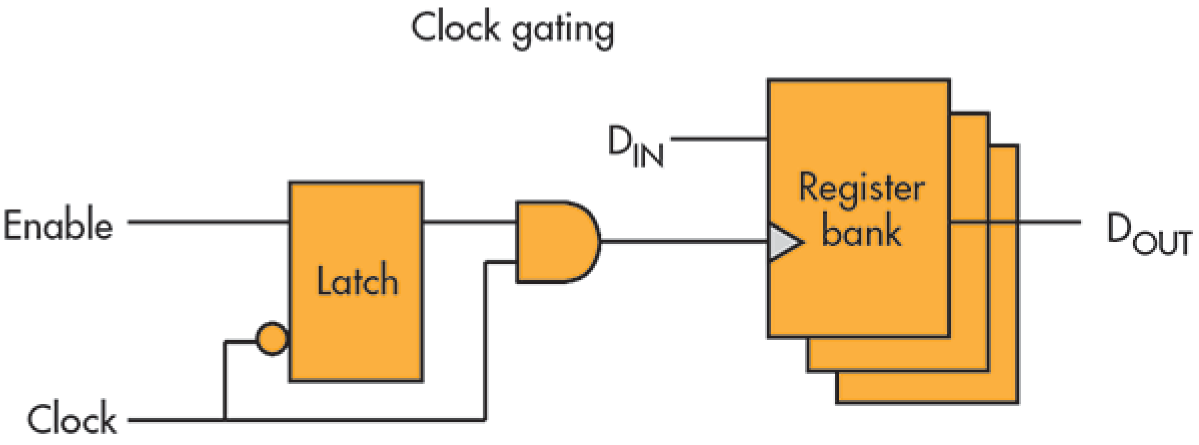
When Clock Gating Falls Short, Turn to Power Gating

Because small geometry processes have high leakage current in relation to dynamic current, clock-gating techniques are proving insufficient in many idle-power-sensitive designs, even when the PHY is in an L1 sub-state. The solution, then, is power gating—literally turning off the power to unused logic to yield true zero-power consumption for portions of the SoC not in use. This is generally done with on-die FET switches controlled by power-management logic on the SoC.

Recall that a PCI or PCI Express device in the “D3” state could have its power removed entirely since the operating system would completely reconfigure the device once power was restored. Likewise, a PCI Express device in the L2 state could operate on auxiliary power and, depending on the protocol, perform a full re-initialization upon return to the normal L0 operating state. Power gating in either of these states would be fairly simple: Turn off the entire chip with the exception of a small block of logic to restore power upon reset, should the system NOT actually remove the external power supply.

Power gating becomes a bit more complex with L1 sub-states, because a main goal of such an effort is to ensure that software needn't reconfigure the device upon return to normal operation. This means that a power-gated SoC has to retain key PCI Express state information, such as its RequesterID, BAR memory mapping, link controls, etc. Even inside the PCI Express PHY, various configuration parameters must be retained to ensure the PHY itself can resume operation quickly (keeping to the 100-µs industry target) without needing reconfiguration from the PCI Express controller logic.

There are two options for preserving this state across power removal: Provide a separate power plane (or “power island”) for that logic, or include “retention cells,” which are flip-flops that save their state upon removal of the primary power source. Of course, retention cells generally rely on a separate power plane—the distinction is more in the design implementation flow. Confusingly, a half-dozen or so variants of retention cells differ in the method by which they're instructed to transfer state to/from retention. Designers must also provide ation cells between powered and unpowered logic to prevent power leakage and potential device damage n floating inputs on powered logic and driving unpowered logic.



L1 sub-states would seem to be an excellent choice for power gating, since the required retained state is well-specified, and there's a simple digital signal ("CLKREQ#") to initiate device wakeup. Unfortunately, the complexity noted above has previously limited power-gating implementations to SoCs where a very tight integration existed between the design team making the PCI Express controller and the implementation team performing the place and route.

The Rise of IEEE 1801

The need to specify retention-cell usage and control, power islands, and isolation-cell placement drove the design implementation industry to develop and adopt the IEEE 1801 standard—more commonly known as the Unified Power Format (UPF). UPF provides a standard for design and verification of low-power integrated circuits that applies across multiple tool vendors. It's a single set of commands to specify power intent to be used in the entire implementation flow. By delivering UPF information to the implementation team, a design team can provide data about voltage supplies, power switches, retention, and isolation, all in a standard format usable by a wide variety of tools.

For the digital logic of a PCI Express controller, the designers can deliver UPF files to specify exactly which modules need state retention, which signals require isolation, and where power switches need to be included. Additional design work provides control logic for power transitions and integrates that logic with the PCI Express L1 sub-states protocol.

Verifying the UPF-driven logic requires some extra effort as well, because the controller design team needs to simulate with tools that accurately reflect the behavior of powered-off digital logic so that any missing isolation cells can be detected before silicon. Furthermore, the variety of UPF implementation types (e.g., the many variants of retention cells mentioned earlier) may require additional effort in verification.

For a PCI Express controller, the configuration space registers are a large portion of the logic that must be retained. That means the exact savings will vary, but 75%-85% is a reasonable number to expect in a typical design.

When to Implement Power Gating

If we consider the example PCIe SSD mentioned above, in L1.1 the x4 PCIe PHY might dissipate between 1 and 1.2 mW of power and the PCIe controller logic another 1 to 2 mW. Therefore, power gating the controller starts to become attractive. In L1.2, though, the PHY only leaks about 40 µW, so the controller's 1 to 2 mW is clearly larger power factor! Cutting 80% of that leakage current brings the controller down to only 200 to 400 µW ›verall idle power.

It should be clear, then, that by supporting L1 sub-states and using UPF to implement power gating, an SoC can implement PCI Express with idle power levels that are quite conducive to battery operation.

PCI Express 3.1 Solutions

Designers have integrated Synopsys' low-power PCI Express 3.1 IP PHYs and controllers in well over 1000 designs due to their low-power features. These include L1 sub-states and use of power gating, power islands, and retention cells, which together cut standby power to less than 10 µW/lane.[1]

In addition, support for supply under drive, a novel transmitter design, and equalization bypass schemes reduce active power consumption to less than 5 mW/Gb/lane, all while meeting the PCI Express 3.1 electrical specification. Together, these features are enabling designers to integrate high-performance PCI Express interfaces into lower-power applications such as mobile devices. Learn more about Synopsys DesignWare IP for PCIe at *www.synopsys.com/pcie*.

Reference:

1. "Synopsys Announces Industry's Lowest Power PCI Express 3.1 IP Solution for Mobile SoCs," Synopsys press release, 2015.

Source URL: http://electronicdesign.com/eda/power-gating-reducing-pcie-power-consumption-mobile-levels